# Transfer Entropy and Applications to Ecosystem Atmospheric Data

Akhilleus Sironen

University of Helsinki
Mathematics and statistics

Master's thesis

June 18, 2020

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta/Osasto — Fakultet/Sektion — Faculty | Laitos — Institution — Department |
|---|---|
| Faculty of Science | Department of Mathematics and Statistics |

Tekijä — Författare — Author
Akhilleus Sironen

Työn nimi — Arbetets titel — Title

Transfer Entropy and Applications to Ecosystem Atmospheric Data

Oppiaine — Läroämne — Subject
Mathematics

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| Master's thesis | June 2020 | 43 |

Tiivistelmä — Referat — Abstract

This thesis studies methods for inferring time series influences on one another. Specifically *transfer entropy* is motivated, derived and explained through *information theory* and Markov chains. Issues and solutions for the calculation of transfer entropy in the context of discrete, coarse grained and continuous time series are covered. The equality of *Granger causality* and transfer entropy in the Gaussian case is explored. Software package TISEAN for *nonlinear time series analysis* is explored and its capabilities are explained.

In this thesis nonlinear methods are applied to measurements from *SMEAR II* station which is located in Hyytiälä Forestry Field Station in Hyytiälä, Finland. The problems caused by nonstationarity of the data are discussed and a solution to these problems is attempted.

The ecosystem exchange is found to be *nonstationary* and containing two different strong periodic components implying that the underlying dynamical system changes its characteristics. This prevented obtaining a phase space embedding from the measured data within the scope of this thesis.

Further work is needed to find out whether transfer entropy can give new insights into analyzing the ecosystem and its influences. Rudimentary ground work has been laid with this thesis and possible ways of mitigating the nonstationarity in the future are given. All the code for reproducing the analysis and plots are included in the appendix to help future work.

Avainsanat — Nyckelord — Keywords
Transfer entropy, Granger causality, nonlinear time series analysis

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula Campus Library

Muita tietoja — Övriga uppgifter — Additional information

# Contents

# Chapter 1

# Introduction

Transfer entropy is a method of analyzing the possible effect a time series has had on another time series using information theoretic principles and the generalized Markov property. My thesis discusses this and the relevant background and attempts to use these methods to analyze ecosystem atmospheric data.

## 1.1 Overview

This thesis consists of several chapters. First the introduction will cover what this thesis is about as well as the general structure. The introduction will also explain some notation used in this thesis.

The main portion of the thesis is in the Methods chapter in which I discuss different methods and theoretical background for these. I will also compare these methods and present an existing software package meant for nonlinear time series analysis.

After the theory chapter I have a short chapter on the data that these methods were used on.

In the penultimate chapter I will describe the work that was done with these methods to analyze the data as well as different problems that were encountered.

In the final chapter I give conclusions and possible future avenues to pursue related to these methods and this or similar data.

Computer code for producing the analyses and plots are included in the appendix.

## 1.2 Notation

I use Granger's [4] notation for the set of past values of a stochastic process and use it for time series in general.

**Definition 1.1.** For a stochastic process or a time series $A$ the *set of past values* is

$$\overline{A}_t = \{A_{t-j}, j = 1,2,\ldots,\infty\}\,.$$

The *base of logarithms* in different entropies and informations only changes the units used[2, p.14], and hence these will be mostly dropped. Also the bounds and indexing of $\sum$ are dropped when the context makes these clear.

I adopt the shorthand used by Schreiber [12] when constructing *words* from states $i$ of process $I$.

**Definition 1.2.** The words of length $k$ constructed from states $x$ of process $X$

$$x_n^{(k)} = (x_n,x_{n-1},\ldots,x_{n-k+1}).$$

## 1.3 Acknowledgements

I would like to thank Samuli Launiainen[1] of Natural Resources Institute Finland (LUKE) for providing the data and information relating to earlier work regarding the subject matter of forest ecosystem interactions as well as Tiia Grönholm for offered support. My special thanks I want to extend to Paolo Muratore Ginanneschi [2] for the discussions we've had and instruction I've received from him.

[1] https://www.luke.fi/en/henkilosto/samuli-launiainen/

[2] https://www.helsinki.fi/fi/ihmiset/henkilohaku/paolo-muratore-ginanneschi-9052749

# Chapter 2

# Methods

The most important methods I use in this thesis come from nonlinear time series analysis. I will give short introductions to *Granger causality*, *information theory*, *Markov chains*, *transfer entropy* and *nonlinear time series analysis*. I also point to more detailed explanations for these. After these I give a short introduction to a specialized software package used in analysis, namely *TISEAN*.

## 2.1 Granger causality

The possibility of inferring causal or predictive relationships in time series data is a tempting one. In [4] Granger proposed a testable measure of causality and clarified the terminology in [5]. To get to the definition in [4] Granger first introduces several notations

**Definition 2.1.** Given time series $X_t$ and $Y_t$ the optimum, unbiased predictor of $X_t$ using the set of values $Y_t$ is $P_t(X|Y)$.

**Definition 2.2.** The error series of the prediction is

$$\varepsilon_t(X|Y) = X_t - P_t(X|Y)$$

**Definition 2.3.** The variance of the prediction[1] $X$ given $Y$ is

$$\sigma_G^2(X|Y) = \mathbb{E}\left[(\varepsilon_t(X|Y) - \mathbb{E}\left[\varepsilon_t(X|Y)\right])^2\right]$$

With these definitions Granger [4] gives the definition for causality

[1] In [4] Granger marked this with simply $\sigma^2(X|Y)$, but to avoid confusion with general conditional variance $\sigma_G^2$ is used in this thesis

**Definition 2.4.** $Y_t$ causes $X_t$ with respect to $D_t$ if

$$\sigma_G^2(X|\overline{D}) > \sigma_G^2(X|\overline{D},\overline{Y})$$

This gives the causality with respect to the set $\{D_t, Y_t\}$. The set $D_t$ is taken to contain usually at least $X_t$ and possibly other conditioning data.

In [4] Granger acknowledges that in practice completely optimal predictors are not available as they may be complicated and proposes using linear predictors for their simplicity. He also notes that other criteria besides variance could be used to evaluate the closeness but chooses variance in connection with linear predictors due to its ease of use and interpretation. Given these restrictions Granger opts to use causality in [4] to mean *"linear causality in mean with respect to a specified set D."*

In [5] Granger revisits the notion of causality. Denoting the conditional distribution of $X$ given $D$ by $f(X|D)$ and the corresponding conditional mean by $\mathbb{E}[X|D]$, Granger gives the following definitions for causality[2].

**Definition 2.5.** If

$$f(X_{t+1}|\overline{D}_t,\overline{Y}_t) \neq f(X_{t+1}|\overline{D}_t)$$

then $Y$ is a *prima facie* cause of $X$ with respect to $D$.

**Definition 2.6.** If

$$\mathbb{E}\left[X_{t+1}|\overline{D}_t,\overline{Y}_t\right] \neq \mathbb{E}\left[X_{t+1}|\overline{D}_t\right]$$

then $Y$ is a *prima facie* cause in mean[3] of $X$ with respect to $D$.

Granger touches shortly on the strength of causality [4, p.433] as a function of frequency. The measure of causality that has become the standard was provided later by Geweke in [3].

**Definition 2.7.** Geweke's measure of linear feedback from $Y$ to $X$

$$\mathcal{F}_{Y\Rightarrow X} = \ln\left(\frac{\left|\sigma_G^2(X|\overline{D})\right|}{\left|\sigma_G^2(X|\overline{D},\overline{Y})\right|}\right)$$

This has several pleasing qualities.

[2]Granger also defines $Y$ not causing $X$ when the left and right hand sides in the equations are equal

[3]Here the *in mean* definition is similar to the notion in [4].

**Corollary 2.8.** *The measure is non-negative*

$$\begin{aligned}
&\left|\sigma_G^2(X|\overline{D})\right| \geq \left|\sigma_G^2(X|\overline{D},\overline{Y})\right| \\
&\Leftrightarrow \frac{\left|\sigma_G^2(X|\overline{D})\right|}{\left|\sigma_G^2(X|\overline{D},\overline{Y})\right|} \geq 1 \\
&\Leftrightarrow \ln\left(\frac{\left|\sigma_G^2(X|\overline{D})\right|}{\left|\sigma_G^2(X|\overline{D},\overline{Y})\right|}\right) \geq 0 \\
&\Leftrightarrow \mathcal{F}_{Y\Rightarrow X} \geq 0
\end{aligned}$$

**Corollary 2.9.** *The measure is zero if and only if $Y$ isn't causing $X$ in the Granger sense*

$$\begin{aligned}
&\left|\sigma_G^2(X|\overline{D}|\right) = \left|\sigma_G^2(X|\overline{D},\overline{Y}|\right) \\
&\Leftrightarrow \frac{\left|\sigma_G^2(X|\overline{D})\right|}{\left|\sigma_G^2(X|\overline{D},\overline{Y})\right|} = 1 \\
&\Leftrightarrow \ln\left(\frac{\left|\sigma_G^2(X|\overline{D})\right|}{\left|\sigma_G^2(X|\overline{D},\overline{Y})\right|}\right) = 0 \\
&\Leftrightarrow \mathcal{F}_{Y\Rightarrow X} = 0
\end{aligned}$$

Geweke list additional motivations [3, p.306] for this definition.

To create the distinction between actual causal relationship and the inferred causality terms *Granger-causality* (*G-causality*) and *Granger-causes* (*G-causes*) are often used and will be adopted in this thesis from here on.

## 2.2 Information theory and entropies

Transfer entropy is based on concepts of information theory and I will first go through these quickly. As noted in 1.2 the bases of the logarithms change the units of entropy. Often base 2 is used and this makes the units bits. The notation used for these is based on [12].

*Entropy* measures the uncertainty of a variable ie. amount of information in the variable.

**Definition 2.10.** The *entropy* of a discrete variable $X$ with distribution $p(x)$

$$H_X = -\sum_x p(x)\log p(x).$$

*Kullback entropy* measures the inefficiency ie. how many extra bits are needed, if the density is assumed to be something other than the true distribution.

**Definition 2.11.** *Kullback entropy* for $X$ with assumed distribution $q(x)$ instead of true distribution $p(x)$ is

$$K_X = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

*Conditional entropy* measures the entropy another process $X$ conditioned on $Y$ brings to the joint entropy.

**Definition 2.12.** *Conditional entropy* of $X$ given $Y$ is

$$H_{X|Y} = -\sum_{x,y} p(x,y) \log p(x|y).$$

These can be viewed as somewhat of building blocks as different measures can be constructed using these. Kullback entropy can be adapted to conditional entropy to arrive at conditional Kullback entropy.

**Definition 2.13.** For true distribution $p(x|y)$ and assumed distribution $q(x|y)$ the *conditional Kullback entropy* is

$$K_{X|Y} = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{q(x|y)}$$

*Mutual information* is a measure of how much more information is needed to describe two processes $X$ and $Y$ as independent with distributions $p_X(x)$ and $p_Y(y)$ if their true distribution is $p_{XY}(x,y)$. This can be seen as applying Kullback entropy to processes $X$ and $Y$ with these distributions.

**Definition 2.14.** *Mutual information* for processes $X$ and $Y$ is

$$M_{XY} = \sum_{x,y} p(x,y) \log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}.$$

## 2.3 Motivation for transfer entropy

In his paper *Schreiber*[12] describes shortcomings of different information theoretical concepts for analyzing flow of information between processes and introduces

a *transfer entropy* to overcome these. I give a quick run-through of the concepts Schreiber analyzes and their perceived shortcomings.

*Mutual information* is symmetric with respect to $X$ and $Y$. On the other hand, it is clear that there are cases in which the flow of information is asymmetrical. Hence mutual information is not suitable for describing flow of information.

*Conditional information* $H_{X|Y} = H_{XY} - H_Y$ is asymmetric with respect to $X$ and $Y$. However

$$\begin{aligned} H_{X|Y} - H_{Y|X} &= H_{XY} - H_Y - (H_{YX} - H_X) \\ &= H_{XY} - H_Y - H_{YX} + H_X \\ &= H_X - H_Y + H_{XY} - H_{YX} \\ &= H_X - H_Y \end{aligned}$$

and so the differences in conditional information are only due to different entropies in $X$ and $Y$ instead of information exchange between processes.

*Time-delayed mutual information* is constructed by adding a time lag $\tau$ to either process $X$ or $Y$, and thus considering these processes at different times.

**Definition 2.15.** *Time-delayed mutual information* for processes $X$ and $Y$ with delay $\tau$ is

$$M_{XY}(\tau) = \sum_n p(x_n, y_{n-\tau}) \log \frac{p_{XY}(x_n, y_{n-\tau})}{p_X(x_n) p_Y(y_n)}.$$

This is asymmetric with relation to $X$ and $Y$ and gives time directionality. Schreiber however calls this an *ad hoc* way, and proposes considering transition probabilities instead of static probabilities as a way of bringing the dynamical structure to the analysis.

## 2.4 Markov chains

To formalize handling of transition probabilities I'll discuss the *Markov property* and *Markov chains*. In short Markov property means that the future of a process is independent of its past beyond the present value ie. the process is memoryless. To define this in more exact terms let $S$ be a *state space* and $X$ be a sequence $X = (x_n \in S : n \geq 0)$ in the state space $S$ as in [6, p.205].

**Definition 2.16.** The *Markov property* is

$$p(x_{n+1} |\, x_n, \ldots, x_0) = p(x_{n+1} |\, x_n), \quad \forall n \geq 0.$$

**Definition 2.17.** A sequence $X$ that satisfies the Markov property is called a *Markov chain.*

In the context of time series analysis there is often a need to be able to work with more than the current state. The Markov property can be generalized into *Markov property of order k* where the state $X_{n+1}$ is dependent only on $k$ last states.

**Definition 2.18.** *Markov property of order k*

$$p(x_{n+1}|x_n, \ldots, x_{n-(k-1)}, \ldots, x_0) = p(x_{n+1}|x_n, \ldots, x_{n-(k-1)}), \quad x_m = x_0, m < 0.$$

For a Markov chain $X$ of order $k$ with states $x \in S$ another process $Z$ in state space $T = S^k$ with $z_n = (x_n, \ldots, x_{n-(k-1)})$ can be constructed. Since $p(x_{n+1})$ is defined by the $k$ preceding states of $X$ which are included in $Z_n$

$$p(z_{n+1}|z_n, \ldots, z_0) = p(z_{n+1}|z_n)$$

and hence $Z$ is a Markov chain. As noted in 1.2 the short-hand $x_n^{(k)}$ is used to denote these sequences $(x_n, \ldots, x_{n-(k-1)})$.

As the probability of state $x_{n+1}$ depends only on $x_n$ when $X$ is a Markov chain these can be incorporated into a *transition matrix.*

**Definition 2.19.** Given a process $X$ the *transition matrix* is

$$T_X(n) = (p_{i,j}(n) : i,j \in S)$$

where the elements are given by the transition probabilities

$$p_{i,j}(n) = P(x_{n+1} = j | x_n = i).$$

A Markov chain is *homogenous* if the transition probabilities do not depend on $n$. In this case the transition matrix simplifies to

$$T_X(n) = T_X(0) = (p_{i,j}(0)) = T_X$$

For a more in depth description of Markov chains and related concepts see eg. [6] and [11].

## 2.5 Entropy rate

*Entropy rate* tells how many bits are needed for one additional state of $X$ if all previous $k$ states are known

$$h_X = -\sum p(x_{n+1}, x_n^{(k)}) \log p(x_{n+1}|x_n^{(k)})$$

As Schreiber points out

$$p(x_{n+1}|x_n^{(k)}) = \frac{p(x_{n+1},x_n^{(k)})}{p(x_n^{(k)})} = \frac{p(x_{n+1},x_n,\ldots,x_{n-k+1})}{p(x_n^{(k)})} = \frac{p(x_{n+1}^{(k+1)})}{p(x_n^{(k)})}$$

and hence the entropy rate can be written as

$$\begin{aligned} h_X &= -\sum p(x_{n+1},x_n^{(k)}) \log p(x_{n+1}|x_n^{(k)}) = -\sum p(x_{n+1}^{(k+1)}) \log \frac{p(x_{n+1}^{(k+1)})}{p(x_n^{(k)})} \\ &= -\sum p(x_{n+1}^{(k+1)}) \left( \log p(x_{n+1}^{(k+1)}) - \log p(x_n^{(k)}) \right) \\ &= -\sum p(x_{n+1}^{(k+1)}) \log p(x_{n+1}^{(k+1)}) - \left( -\sum p(x_{n+1}^{(k+1)}) \log p(x_n^{(k)}) \right) \\ &= H_{X^{(k+1)}} - H_{X^{(k)}} \end{aligned}$$

where $H_{X^{(k)}}$ is the Shannon entropy for $k$ dimensional delay vectors.

For more reading on entropy rate see eg. [2, chapter 4.2]

## 2.6 Transfer entropy

With these tools and knowledge Schreiber [12] starts defining transfer entropy. The basis of the definition stems from two main ideas:

- Entropy is based on the static probabilities of states whereas entropy rate is based on the transition probabilities, which describe the dynamics of the processes.
- If there is no information flow from $Y$ to $X$, the transition probabilities of $X$ are not changed even when considering the history of $Y$. The deviation from this assumption can be measured using Kullback entropy.

**Definition 2.20.** *Transfer entropy* is

$$T_{Y\to X} = \sum p(x_{n+1}, x_n^{(k)}, y_n^{(l)}) \log \frac{p(x_{n+1}|x_n^{(k)}, y_n^{(l)})}{p(x_{n+1}|x_n^{(k)})}. \tag{2.21}$$

Now $T_{Y\to X}$ is not symmetric and measures explicitly the dependence of transition probabilities of $X$ on $Y$. It is possible to also exclude other factors $D$ by incorporating these in the condition ie.

$$T_{Y\to X|D} = \sum p(x_{n+1}, x_n^{(k)}, y_n^{(l)}, d_n^{(m)}) \log \frac{p(x_{n+1}|x_n^{(k)}, y_n^{(l)}, d_n^{(m)})}{p(x_{n+1}|x_n^{(k)}, d_n^{(m)})}.$$

With the definition of transfer entropy it is time to look at computing it in different cases.

### 2.6.1 Discrete data

For discrete data the calculation of transfer entropy is straightforward. The joint probabilities are easily-defined and the conditional probabilities are easily transformed to joint probabilities with

$$p(x|y) = \frac{p(x,y)}{p(y)}.$$

### 2.6.2 Coarse grained data

When the underlying data $(X,Y)$ is continuous but is coarse grained[4] the partitioning of state space must be considered. Schreiber points out in [12] that when the system is coarse grained at resolution $r$ the limit

$$\lim_{r\to 0} T_{Y\to X}(r)$$

is independent of the partition and finite unless $X$ and $Y$ are deterministically coupled. Usually, however, the limit $r \to 0$ is impossible to reach. Schreiber offers as solutions either fixing the resolution or studying the transfer entropy as a function of the resolution.

### 2.6.3 Continuous time series

Without coarse graining the data, the state space is continuous. Adapting concepts of nonlinear time series analysis Schreiber [12] proposes estimating the joint probabilities using kernel estimation over all available realizations of $(x_{n+1}, x_n^{(k)}, y_n^{(l)})$ in the time series.

**Definition 2.22.** *Kernel estimate* of the joint probability is[5]

$$\hat{p}_r(x_{n+1}, x_n, y_n) = \frac{1}{N}\sum_{n'} K\left(r - \left|\begin{pmatrix} x_{n+1} - x_{n'+1} \\ x_n - x_{n'} \\ y_n - y_{n'} \end{pmatrix}\right|\right).$$

Schreiber uses the step kernel

$$\Theta(u) = \begin{cases} 0, & u \leq 0 \\ 1, & u > 0 \end{cases}$$

and maximum distance as the $|\cdot|$ norm, but notes that other options can be considered.

[4] ie. the state space is discretized

[5] In Schreiber [12] the order of the subtraction is wrong

## 2.7 Comparison of Granger causality and Transfer entropy

Both Granger causality and transfer entropy are ways of measuring the impact of a time series to another, and it is natural to ask what is their relationship to each other. Barnett, Barrett and Seth proved in [1] that these are equivalent for Gaussian variables.

With $X$ and $Y$ being jointly distributed random vectors the authors consider the linear regression

$$X = \alpha + Y \cdot A + \varepsilon \tag{2.23}$$

with constants $\alpha$, regression coefficients $A$ and residuals $\varepsilon$. Taking covariance of equation (2.23) gives

$$\begin{aligned}
\Sigma(X,X) &= \Sigma(\alpha + Y \cdot A + \varepsilon, \alpha + Y \cdot A + \varepsilon) \\
\Sigma(X) &= \Sigma(Y \cdot A, Y \cdot A + \varepsilon) + \Sigma(\varepsilon, Y \cdot A + \varepsilon) \\
\Sigma(X) &= \Sigma(Y \cdot A, Y \cdot A) + \Sigma(Y \cdot A, \varepsilon) + \Sigma(\varepsilon, Y \cdot A) + \Sigma(\varepsilon, \varepsilon)
\end{aligned}$$

and when $\varepsilon$ and $Y$ are uncorrelated, further

$$\Sigma(X) = A \cdot \Sigma(Y) \cdot A^T + \Sigma(\varepsilon) \tag{2.24}$$

On the other hand

$$\begin{aligned}
0 &= \Sigma(Y,\varepsilon) \\
&= \Sigma(Y, X - \alpha - Y \cdot A) \\
&= \Sigma(Y, X) - \Sigma(Y, Y \cdot A) \\
&= \Sigma(X, Y)^T - \Sigma(Y, Y) \cdot A \\
&= \Sigma(X,Y)^T - \Sigma(Y) \cdot A \\
\Rightarrow \quad \Sigma(X,Y)^T &= \Sigma(Y) \cdot A \\
\Rightarrow \quad \Sigma(Y)^{-1} \cdot \Sigma(X,Y)^T &= A
\end{aligned}$$

Combining this with equation (2.24)

$$\begin{aligned}
\Sigma(X) &= A^T \cdot \Sigma(Y) \cdot A + \Sigma(\varepsilon) \\
\Sigma(\varepsilon) &= \Sigma(X) - A^T \cdot \Sigma(Y) \cdot A \\
\Sigma(\varepsilon) &= \Sigma(X) - (\Sigma(Y)^{-1} \cdot \Sigma(X,Y)^T)^T \cdot \Sigma(Y) \cdot \Sigma(Y)^{-1} \cdot \Sigma(X,Y)^T \\
\Sigma(\varepsilon) &= \Sigma(X) - \Sigma(X,Y) \cdot (\Sigma(Y)^{-1})^T \cdot \Sigma(Y) \cdot \Sigma(Y)^{-1} \cdot \Sigma(X,Y)^T \\
\Sigma(\varepsilon) &= \Sigma(X) - \Sigma(X,Y) \cdot \Sigma(Y)^{-1} \cdot \Sigma(X,Y)^T \\
\Sigma(\varepsilon) &= \Sigma(X|Y).
\end{aligned}$$

The authors then consider two regression models for three processes $X$, $Y$, $Z$

$$\begin{aligned}
x_t &= \alpha_t + (x_{t-1}^{(p)} \oplus z_{t-1}^r) \cdot A + \varepsilon_t \\
x_t &= \alpha_t' + (x_{t-1}^{(p)} \oplus y_{t-1}^q \oplus z_{t-1}^r) \cdot A' + \varepsilon_t'
\end{aligned}$$

and use Geweke's measure of G causality to arrive at

$$\begin{aligned}
\mathcal{F}_{Y \Rightarrow X|Z} &= \ln \left( \frac{|\Sigma(\varepsilon_t)|}{|\Sigma(\varepsilon_t')|} \right) \\
&= \ln \left( \frac{\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus z_{t-1}^{(r)})\right|}{\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus y_{t-1}^{(q)} \oplus z_{t-1}^{(r)})\right|} \right).
\end{aligned} \tag{2.25}$$

The authors define transfer entropy by differences of entropies of $X$ conditioned on its own past and $Z$ against $X$ conditioned on its own past and $Y$ and $Z$

$$\mathcal{T}_{Y \to X|Z} = H(x_t|x_{t-1}^{(p)} \oplus z_{t-1}^{(r)}) - H(x_t|x_{t-1}^{(p)} \oplus y_{t-1}^{(q)} \oplus z_{t-1}^{(r)}).$$

The authors then extend the expression for entropy of multivariate Gaussian variables of dimension $n$

$$H(X) = \frac{1}{2} \ln (|\Sigma(X)|) + \frac{1}{2} n \ln(2\pi e)$$

by showing that conditional entropy is

$$H(X|Y) = \frac{1}{2} \ln (|\Sigma(X|Y)|) + \frac{1}{2} n \ln(2\pi e).$$

With these transfer entropy for Gaussian variables $X$, $Y$, $Z$ becomes

$$\begin{aligned}\mathcal{T}_{Y\to X|Z} =& H(x_t|x_{t-1}^{(p)} \oplus z_{t-1}^{(r)}) - H((x_t|x_{t-1}^{(p)} \oplus y_{t-1}^{(q)} \oplus z_{t-1}^{(r)}) \\ =& \frac{1}{2}\ln\left(\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus z_{t-1}^{(r)})\right|\right) + \frac{1}{2}n\ln(2\pi e) \\ & - \left(\frac{1}{2}\ln\left(\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus y_{t-1}^{(q)} \oplus z_{t-1}^{(r)})\right|\right) + \frac{1}{2}n\ln(2\pi e)\right) \\ =& \frac{1}{2}\ln\left(\frac{\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus z_{t-1}^{(r)})\right|}{\left|\Sigma(x_t|x_{t-1}^{(p)} \oplus y_{t-1}^{(q)} \oplus z_{t-1}^{(r)})\right|}\right). \end{aligned} \tag{2.26}$$

Combining now equations for Granger causality (2.25) and transfer entropy (2.26) Barnett, Barrett and Seth [1] arrive at

$$\mathcal{F}_{Y\Rightarrow X|Z} = 2\mathcal{T}_{Y\Rightarrow X|Z}.$$

## 2.8 Nonlinear time series analysis

The term *nonlinear* in nonlinear time series analysis is used to separate the methods from the more usual (linear) time series analysis, in which the structure of the data set is interpreted through linear correlations. This in turn means that any *irregularities* in the time series have to be attributed to an external irregular source. [9, p. 3] From dynamical system research it is known however that nonlinear dynamical systems can also cause chaotic and irregular behaviour even when there are no external inputs to the system [8]. For more in depth reading regarding nonlinear time series analysis see eg. [9].

## 2.9 TISEAN package

As the computer implementations for basic nonlinear methods I use the package *TISEAN* [7] specifically version 3.0.1. It is a GPL licensed software package containing methods based on theory of nonlinear deterministic dynamical systems. The name TISEAN comes from **ti**me **se**ries **an**alysis.

As described in their article [7] the main design philosophy of the package has been to avoid black boxes ie. programs where one were to throw data in and get a single value back. Instead the programs compute new variables relevant to the analysis to interpret. While TISEAN contains several linear time series analysis tools, they are meant for a quick inspection of data and not to replace tools

specifically aimed at linear time series analysis. These programs are command line driven and support piping.

As the TISEAN package is focused on nonlinear deterministic dynamical system analysis, many of the algoritms work in multidimensional phase space. Time series on the other hand are (usually scalar) sequences $s_n = s(x_n)$. As such the time series needs to be unfolded to an embedding space. TISEAN offers several tools that help in this. `delay` is a tool to just convert time series into a multidimensional time series given an embedding dimension $m$ and lag $\tau$. TISEAN contains several programs to help decide on the embedding parameters:

- `false_nearest` calculates false nearest neighbors to help decide on an embedding dimension
- `corr`, `autocorr` and `mutual` can be used to decide on the lag

Besides these TISEAN offers programs for mapping the data to a lower dimension by principal component analysis (`pc`, `pca`) and to down sample data by taking Poincaré intersections (`poincare`,`extrema`).

As methods work in phase space calculating local quantities requires finding the points in the neighborhood. A naive search for neighbors is $O(N^2)$ which is computationally prohibitive for larger $N$. There are several methods for finding neighbors more effectively. TISEAN uses a box-assisted method, where by candidates for neighbors are defined through a two dimensional grid. All the neighbors must then be located in adjacent boxes, though not all points in the adjacent boxes are neighbors.

When calculating Lyapunov exponents or estimates of entropy and dimension the ensemble averages are often replaced by time averages. If the data is not stationary, the results will be wrong. As a way to visualize possible non-stationarity TISEAN offers `recurr` that allows creation of recurrence plots. If the points used for calculations are too much temporally correlated this will introduce a bias as well. This can be visualized by a space-time separation plot for which TISEAN offers `stp`.

TISEAN offers several methods for prediction of time series. Version 3.0.1 of TISEAN has methods for locally zeroth order predictor `lzo-run` and locally linear predictor `lfo-run` as well as global predictors using polynomials (`polynom`) or radial basis functions (`rbf`). The package also contains programs for nonlinear noise reduction, since linear filters might interfere with the nonlinear structure.

TISEAN also contains multiple methods for calculating Lyapunov exponents. Estimation of the maximum Lyapunov exponent can be done with `lyap_k` or `lyap_r`). All the Lyapunov exponents can be estimated using `lyap\_spec`). It must be noted that the number of Lyapunov exponents is equal to the dimension of phase space there is a possibility of spurious Lyapunov exponents as the original

phase space dimension is unknown and the analysis is done in embedding space where the dimension might differ. TISEAN also has multiple methods for calculating and estimating dimensions. These include `c1` for information dimension and `d2` for correlation sums, dimension, and entropy as well as `boxcount` for estimating Renyi entropies.

# Chapter 3

# Ecosystem atmospheric data

In this thesis I will be using aforementioned methods in analysing ecosystem atmospheric data. The original plan was to analyze the ecosystem exchange using Granger causality and transfer entropy and compare the results these gave.

## 3.1 Description of data

The data consists of time series for different physical attributes relevant to the micrometeorological and ecological state of a forest. The measurements are from *SMEAR II* station located in Hyytiälä, Finland. This station is part of Hyytiälä Forestry Field Station.

The data had been preprocessed by Samuli Launiainen before this thesis was started and was separated to two different files with the following variables:

- `FIHy_flx_1997-2017.dat` containing ecosystem-atmosphere fluxes
- `FIHY_forcing_1997-2017.dat` containing meteorological measurements as well as other variables that could be considered as forcings

Both of these files span the years 1997 through to 2017 with 30 minute intervals (368160 samples) and have the same first 6 columns describing the time:

1. *year*
2. *month*
3. *day*
4. *hour*
5. *min*: minute part of the time
6. *doy*: day of year

### Fluxes

The ecosystem-atmosphere fluxes contain measurements for how the ecosystem is interacting with the atmosphere. This data contains also some quality control flags

7. $NEE$: net ecosystem exchange ie. the $CO_2$ balance; ($\mu mol\, m^{-2}\, s^{-1}$ )
8. $GPP$: gross primary productivity ie. the $CO_2$ uptake; ($\mu mol\, m^{-2}\, s^{-1}$ )
9. $R_{eco}$: ecosystem respiration ie. the $CO_2$ release; ($\mu mol\, m^{-2}\, s^{-1}$ )
10. $H$: sensible heat flux; $W\, m^{-2}$
11. $G_{flux}$: Ground heat flux; $W\, m^{-2}$
12. $R_{net}$: net radiation balance; $W\, m^{-2}$
13. $Qc_{NEE}$: quality flag for $NEE$, $GPP$, $R_{eco}$; 0 for observations, $> 0$ for gap-filled values
14. $Qc_H$: quality flag for $H$; 0 for observations, $> 0$ for gap-filled values
15. $LE$: latent heat flux; ($W\, m^{-2}$)
16. $ET$: evapotranspiration; ($\mu mol\, m^{-2}\, s^{-1}$)

and constructed alternative variables for $NEE$, $GPP$ and $R_{eco}$ with different methods for gap-filling and partitioning:

17. $NEE_1$
18. $GPP_1$
19. $GPP_2$
20. $GPP_3$
21. $GPP_4$
22. $R_{eco_1}$
23. $R_{eco_2}$
24. $R_{eco_3}$
25. $R_{eco_4}$
26. $Qc_{NEE_1}$

### Forcings

The forcings contain meteorological and other measurements and factors that can be considered external forcings on the system as a whole.

7. $U$: mean horizontal wind speed; ($m\, s^{-1}$)

8. $u_*$: friction velocity; ($\mathrm{m\,s^{-1}}$)
9. $T_a$: air temperature; (°C)
10. $RH$: relative humidity; (%)
11. $CO_2$: air $CO_2$ mixing ratio; (ppm)
12. $H_2O$: air $H_2O$ mixing ratio; (parts per thousand)
13. $O_3$: air $O_3$ mixing ratio; (ppb)
14. $Prec$: precipitation; (mm per 30 min)
15. $P$: air pressure; (kPa)
16. $PAR_{dir}$: direct photosynthetically active radiation (PAR); ($\mathrm{W\,m^{-2}}$)
17. $PAR_{diff}$: diffuse PAR radiation; ($\mathrm{W\,m^{-2}}$)
18. $NIR_{dir}$: direct near-infrared (NIR) radiation; ($\mathrm{W\,m^{-2}}$)
19. $NIR_{diff}$: diffuse NIR radiation; ($\mathrm{W\,m^{-2}}$)
20. $R_{net}$: net radiation; ($\mathrm{W\,m^{-2}}$)
21. $LW_{in}$: incoming thermal radiation; ($\mathrm{W\,m^{-2}}$)
22. $LW_{out}$: outgoing thermal radiation; ($\mathrm{W\,m^{-2}}$)
23. $LW_{net}$: net thermal radiation; ($\mathrm{W\,m^{-2}}$)
24. $T_{s_h}$: soil temperature in humus layer; (°C)
25. $T_{s_a}$: soil temperature in A-horizon (at depth of $5-10$ cm); (°C)
26. $T_{s_c}$: soil temperature in C-horizon (at depth of 30 cm); (°C)
27. $W_h$: soil moisture in humus; ($\mathrm{m^3\,m^{-3}}$)
28. $emiatm$: atmospheric emissivity, estimated; dimensionless
29. $cloud$: cloud fraction, estimated; dimensionless
30. $REW$: relative plant extractable water in root zone; dimensionless
31. $\Psi_s$: soil water potential in A-horizon, approximative; (MPa)
32. $Zen$: solar zenith angle; (rad)
33. $Azim$: solar azimuth angle; (rad)
34. $Daylength$: length of period when $Zen > 0$; (hours)
35. $W_s$: soil moisture in A-horizon; ($\mathrm{m^3\,m^{-3}}$)
36. $T_{daily}$: daily mean temperature; (°C)
37. $X$: phenology model parameter; (°C)
38. $DD_{sum}$: degree-day sum with $T_0 = 5$°C ie. sum of non-negative values of ($T_{daily} - 5$°C) since the beginning of the year; (°C)

# Chapter 4

# Methods applied to data

In this chapter I'll describe the process of applying different methods to the data and issues that I encountered. Actual code is included and documented in Appendix A.

## 4.1 Preprocessing data

The measured data was in two separate files as described in chapter 3. Before being able to run most tools these files were preprocessed.

Firstly the format of the files was changed slightly. The file containing fluxes used comma (`,`) as the column separator whereas the forcings file used semicolon (`;`). *gnuplot* and *TISEAN* both expect columns to be separated by whitespace and while *gnuplot* can be set to use other separators it was natural to change both data files to use single space (␣) as the column separator.

In the forcings file, columns for $O_3$, $LW_{net}$, $T_{s_h}$ had no measurements and these were removed from the data set.

## 4.2 First analysis

To be able to leverage nonlinear methods the embedding parameters are needed and the data is expected to be stationary. My first steps were to find proper embedding parameters.

### 4.2.1 Finding the lag

In order to create an embedding into phase space a value for lag $\tau$ needs to be decided. If a time series has a periodic component, one possible candidate is to use a quarter of the period([9, p.39]). Since the data shows a strong daily periodic component one possible choice would be to use 6 hours. Because the measurements are done with samples 30 minutes this would give $\tau = 12$. On the other hand the data also has an annual periodic component. As one year has 17520 samples this gives $\tau = 4380$.

Another way to find a suitable $\tau$ is to look at the autocorrelations of the time series. It is desirable that the time lag is not too short, so as not to have the samples too correlated, nor too large, so as not to have the samples being wholly uncorrelated. As such, the first zero of the autocorrelations is one possible candidate.

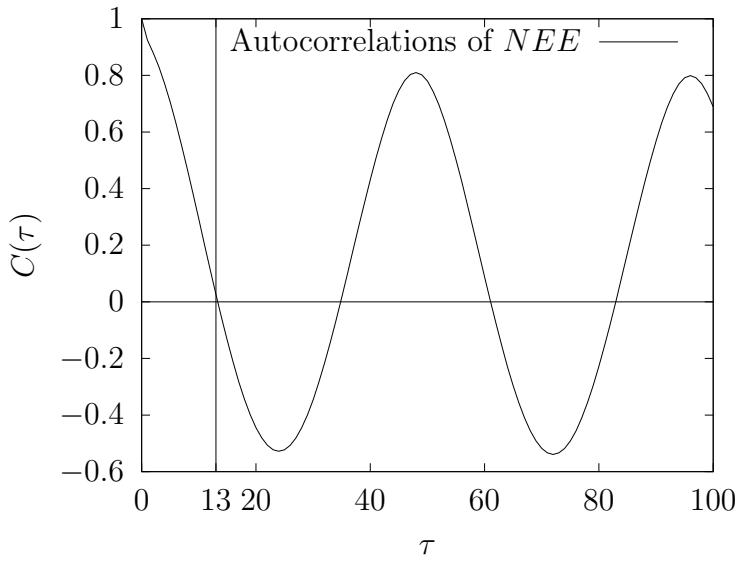


Figure 4.1: Autocorrelations of $NEE$ for $\tau \leq 100$

On the short time frame, selecting $\tau$ based on first zero of autocorrelation would give $\tau = 13$ which is in close agreement with above when selecting $\tau$ as one quarter of the period.

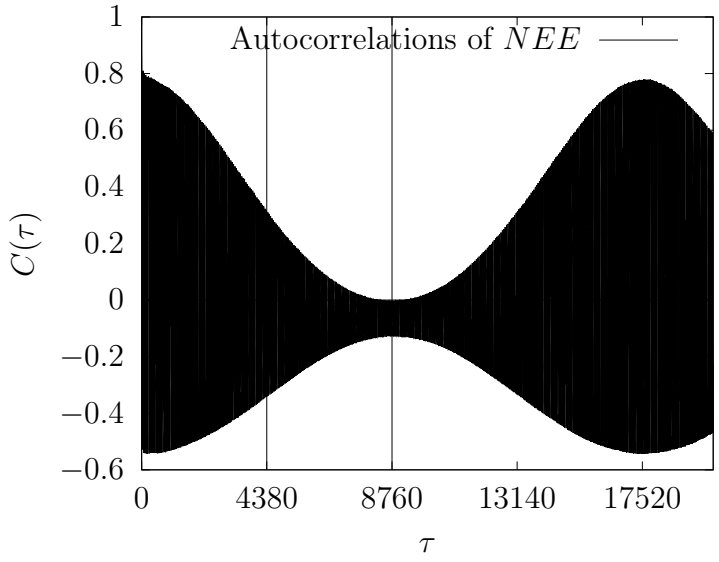


Figure 4.2: Autocorrelations of $NEE$ for $\tau \leq 20000$

Looking at the autocorrelations for a longer lags, it is seen that the variance of autocorrelations decreases until the lag is approximately six months and then increases again.

### 4.2.2 Temporal correlations

To avoid spurious temporal correlations when analyzing the data a space-time separation plot was drawn.

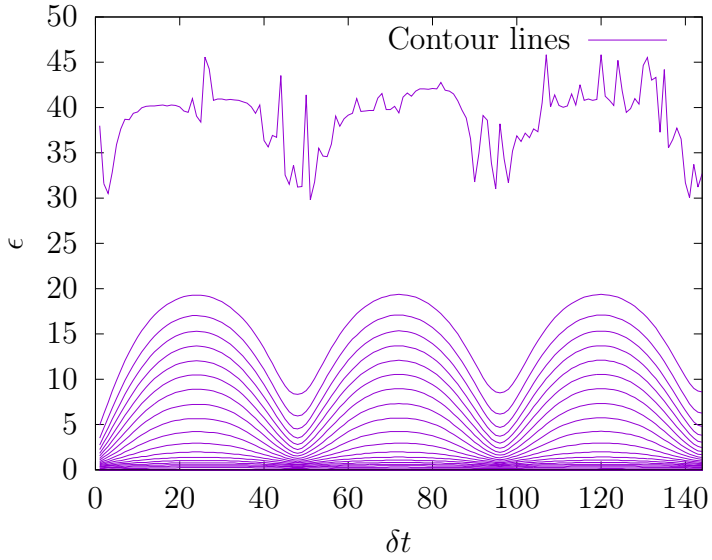


Figure 4.3: Space-time separation plot of $NEE$

As can be seen from figure 4.3 temporal correlations are strong in the beginning and vary cyclically. As the time series is long increasing the Theiler window to leave out temporally correlated samples could be done generously. As such time separation of 480 samples was used as Theiler window when relevant.

### 4.2.3 False nearest neighbors

Embedding the delay vectors in phase space requires also the embedding dimension. For this purpose I ran a false nearest neighbor analysis to see how varying the embedding dimension would alter the false neighbor rate.

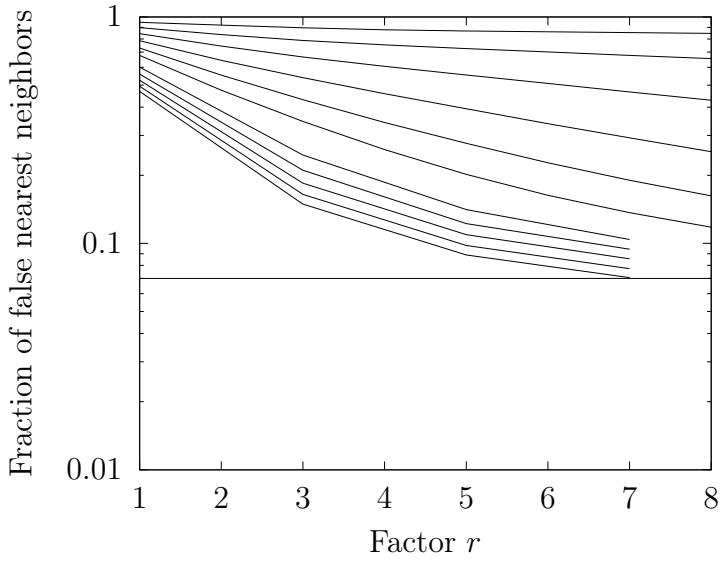


Figure 4.4: False nearest neighbors of $NEE$.

The false nearest neighbor plot was drawn with $\tau = 12$. For embedding dimensions $2 \leq m \leq 7$ integers 1 through 8 were used as the factors $r$, but for embedding dimensions $8 \leq m \leq 12$ only odd integers were used. Even with relatively high embedding dimension of 12 the false nearest neighbor rate remained rather high, approaching ca. 7%.

### 4.2.4 Change in characteristics

When the delay plots were inspected visually, it became apparent that the characteristics of the system change throughout the year. I broke the delay vectors down into groups of 30 days, so that each slice covers approximately one month.

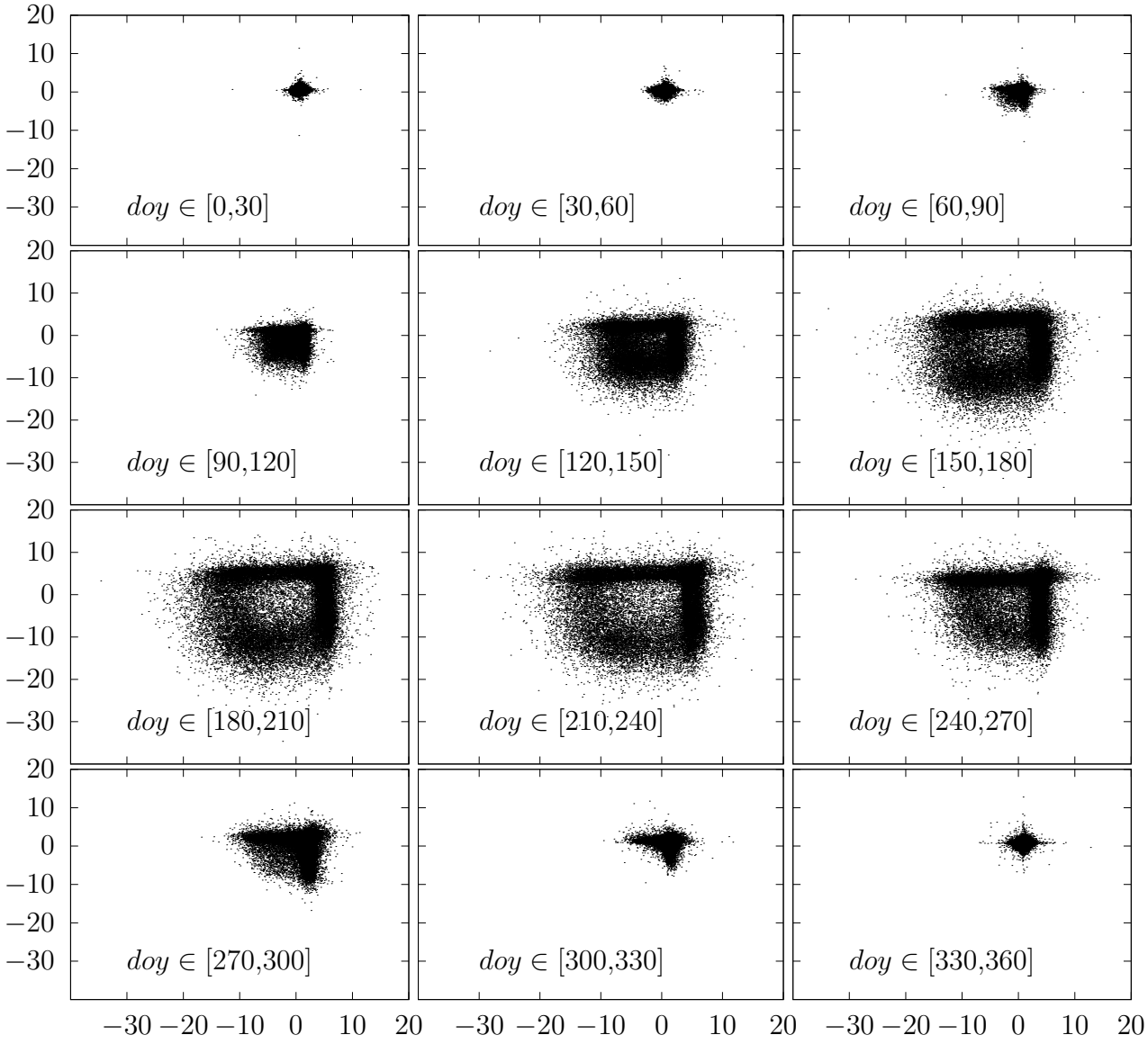


Figure 4.5: Delay graphs of $NEE$ with $\tau = 12$ for 30 day intervals. Each plot contains all years in the data.

From these it is clear that from late autumn to mid-spring ($doy < 90 \vee doy > 300$) the system is not very active and appears as a highly concentrated blob. From summer to early autumn ($150 < doy < 270$) the system is very active and has a more scatter like quality. During this time period the concentrated vertical and horizontal line correspond to the beginning and end of day.

## 4.3 Trying to eliminate nonstationarity

Transfer entropy expects the data to be stationary so as to approximate the transition probabilities as constant over time. The nonstationarity was a problem for the analysis and some steps were taken to make the time series stationary.

The initial idea was to remove the apparent double-cycle from the data by considering only daily averages for the time series. Also the possibility of removing the yearly variation was considered by eg. averaging each 30 minute interval over the years, but this was not pursued further.

### 4.3.1 Daily averages

Both the fluxes and forcings were averaged over each day, but the main focus was kept on $NEE$. With the new daily averages the daily cycle was removed and the sample frequency was changed. A plot of the time series shows clearly that a lot of the high frequency data has been filtered.

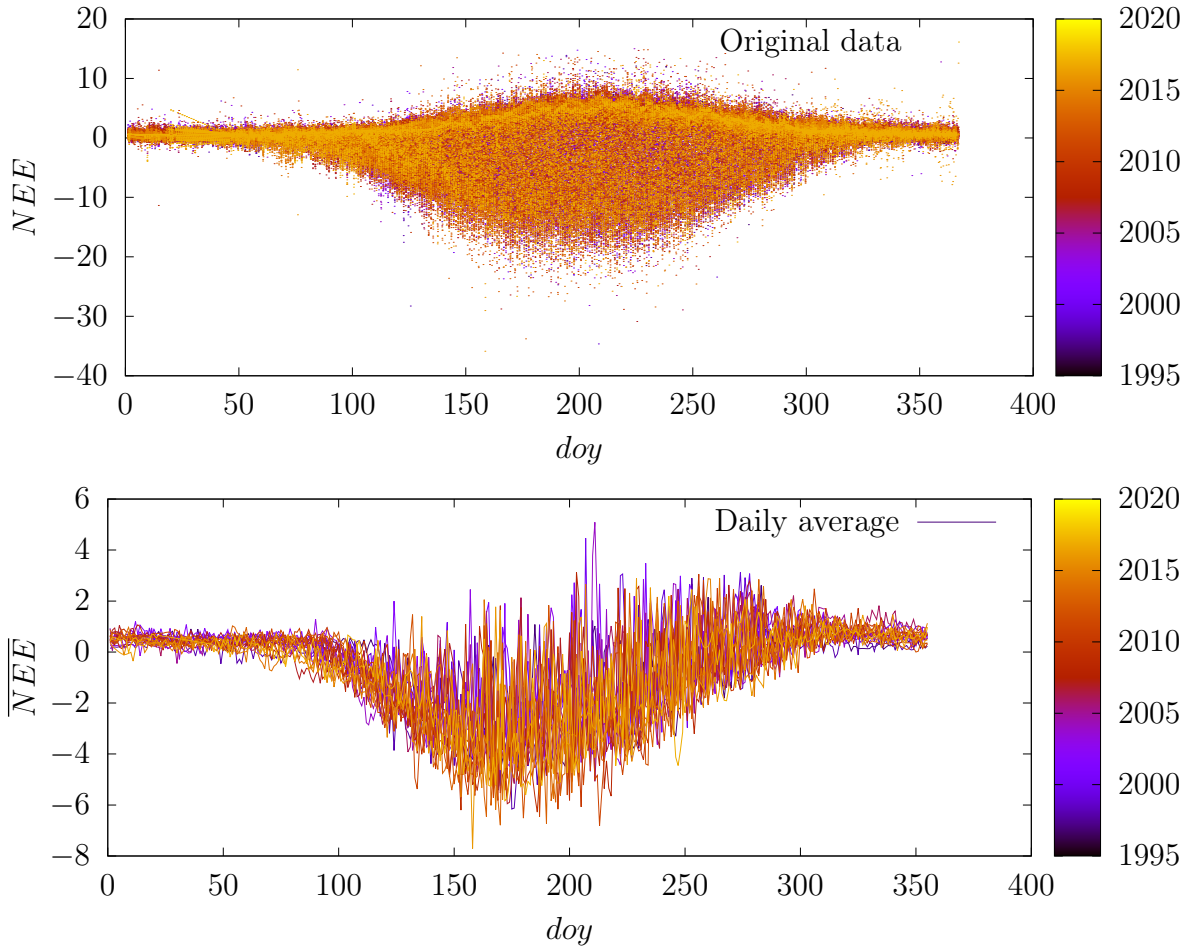


Figure 4.6: $NEE$ and $\overline{NEE}$.

The daily average $\overline{NEE}$ still contains a lot variation from end of spring to beginning of autumn ($120 \leq doy \leq 270$).

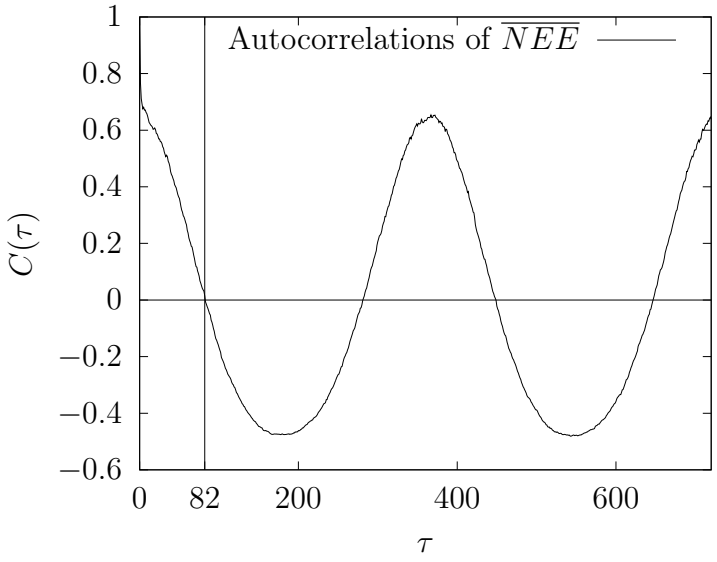


Figure 4.7: Autocorrelations of $\overline{NEE}$ for $\tau \leq 720$.

As the sampling frequency has changed we need to choose a new lag. Selecting lag as quarter of the period would give $\tau = 91$. Selecting first zero of the autocorrelation as lag gives $\tau = 82$. These are again close to each other and I continued working with $\tau = 91$.

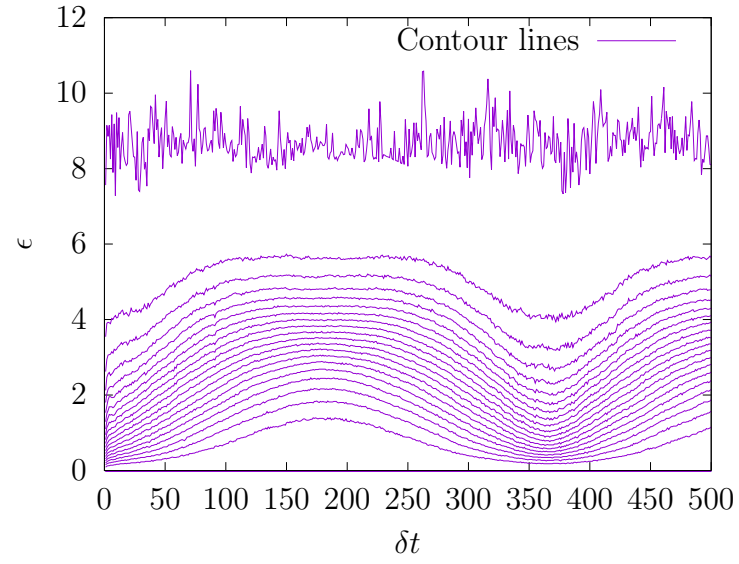


Figure 4.8: Space-time separation plot of $\overline{NEE}$

The space time separation plot shows similar structure as for the original data, but the time scales are different. As could be expected, the yearly cycle is strongly visible. As both the time series length shortening and the period growing by a large factor it the temporal separation can not be taken as 10 cycles as this would discard a significant portion of the data.

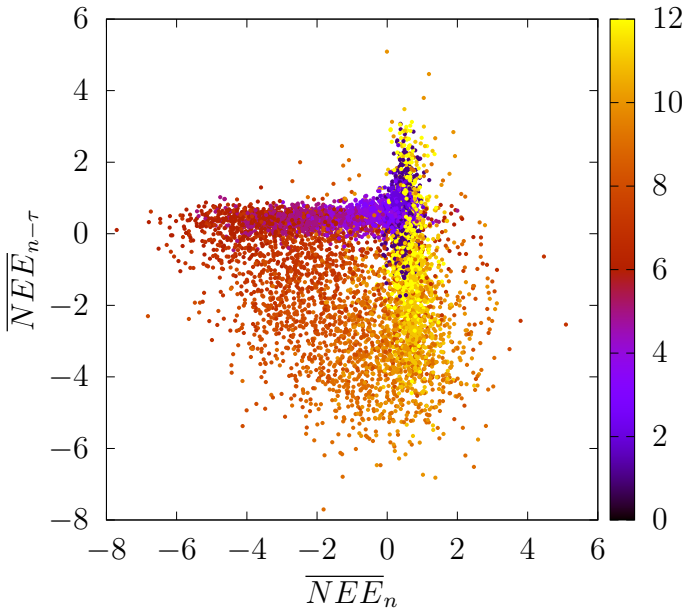


Figure 4.9: Delay graph of $\overline{NEE}$ with $\tau = 91$. Color coding based on month of $\overline{NEE}_n$

Inspecting the delay graph visually reveals how the still remaining large variations from spring to autumn gives the plot a scatter-like quality. The more defined areas correspond to the start of growth season as well as the end of growth season, forming the horizontal and vertical concentrations in the plot.

# Chapter 5

# Conclusions

The original plan was to analyze the ecosystem atmospheric data using transfer entropy and compare it to results with Granger causality. As the data analysis work was started the nonstationarity of the data became apparent and caused problems.

To work with the data using nonlinear methods an embedding using delay coordinates was attempted. Two different methods for selecting the lag were used and both gave similar results. In future analysis of similar data either method can be used, but if the period of a strong periodic components is known a priori using a quarter of the period as the lag allows speeding past this stage.

The second part of embedding the system in a phase space would be to find a suitable embedding dimension. This was attempted using false nearest neighbor method. However it seems that the nonstationarity of the data and the change in the characteristics of the system caused this analysis to consistently show relatively large percentage of neighbors as false. In future analysis more care needs to be put into transforming the data into a stationary series.

Due to time constraints and issues faced the actual transfer entropy analysis could unfortunately not be performed on this data within the scope of this thesis. The current version (3.0.1) of TISEAN package also does not contain a program for computing the transfer entropy and the analysis needs to be performed another tool.

## 5.1 Future work

Further study into this is needed to properly analyze the feasibility of using transfer entropy and other nonlinear methods for analyzing this type of ecosystem atmospheric data. In future studies more care and work needs to be focused especially

in combating nonstationarity. Possible avenues for this include segmenting the data into nearly stationary sections, but special care needs to be taken if trying to analyze the delayed effects of drought or similar effects on the ecosystem to handle cases in which the causes lie outside the segments.

New tools for working with dynamical systems and time series are being created. As transfer entropy calculations are not within the scope of TISEAN some other packages need to be used when these methods can be properly used with the data. A promising relatively new package that could be used is *JIDT* [10], the Java Information Dynamics Toolkit.

# Bibliography


[1] Lionel Barnett, Adam B. Barrett, and Anil K. Seth. Granger causality and transfer entropy are equivalent for gaussian variables. *Phys. Rev. Lett.*, 103:238701, Dec 2009.

[2] T. M. Cover. *Elements of information theory*. Wiley-Interscience, Hoboken, N.J., 2nd ed edition, 2006. Previous ed.: New York: Wiley, 1991.

[3] John Geweke. Measurement of linear dependence and feedback between multiple time series. *Journal of the American Statistical Association*, 77(378):304–313, 1982.

[4] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[5] C.W.J. Granger. Some recent development in a concept of causality. *Journal of Econometrics*, 39(1):199 – 211, 1988.

[6] Geoffrey Grimmett. *Probability : an introduction*. Oxford University Press, Oxford, England, second edition edition, 2014.

[7] Rainer Hegger, Holger Kantz, and Thomas Schreiber. Practical implementation of nonlinear time series methods: The TISEAN package. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9(2):413–435, 1999.

[8] Morris W. Hirsch, Robert L. Devaney, and Stephen Smale. *Differential equations, dynamical systems, and an introduction to chaos /*. Elsevier/Academic Press,, Amsterdam ;, c2004. Rev. ed. of: Differential equations, dynamical systems, and linear algebra / Morris W. Hirsch and Stephen Smale. 1974.

[9] Holger Kantz and Thomas Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 2 edition, 2003.

[10] Joseph T. Lizier. JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI*, 1:11, 2014.

[11] J. R. Norris. *Markov chains*. Cambridge series on statistical and probabilistic mathematics. Cambridge University Press, Cambridge, 1997.

[12] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85:461–4, 08 2000.

# Appendix A

# Computer code

To make rerunning the analysis or performing similar steps on other data files as fluent as possible I tried to emphasize using command line and scripts.

## A.1 Preprocessing

The files were formatted to use the same separator and columns were dropped with the following shell script.

```
echo "Cleaning flux file" #Change input file accordingly
cp FIHy_flx_1997-2017.dat flx.dat
sed -i 's/,/ /g' flx.dat

echo "Cleaning forcing file" #Change input file accordingly
cp FIHy_forcing_1997-2017.dat forcing.dat
sed -i 's/;/ /g' forcing.dat

echo "Removing columns with no data from forcings" #Check if there are columns with no data
cut -d" " -f13,23,24 --complement forcing.dat > fclean.dat  #Similar step can be taken with flux-file
```

## A.2 Combining files

The selected columns from forcings and fluxes files were combined with the following shell script.

```
echo "Selecting promising forcings" #Modify based on findings
cut -d" " -f-6,9,21-22,25-26,34,36,37 forcing.dat > fsel.dat
echo "Selecting basic flux data" #Modify based on findings
cut -d" " -f-9,13 flx.dat > flx_temp.dat

echo "Combining forcings and flux data"
paste -d" " flx_temp.dat fsel.dat > fcomb.dat
rm flx_temp.dat
```

## A.3 Data aggregation

The preprocessed data was aggregated with the following shell script using mostly *GNU* tools `sort`, `join` and `awk`. For the actual aggregation calculation a ready tool was not readily available and these were implemented as `awk` programs described in more details in A.3.1.

```
#Sort by minute-of-year
sort -o flx_sort_moy.dat -k 6,6n -k 4,4n -k 5,5n flx.dat
sort -o forcing_sort_moy.dat -k 6,6n -k 4,4n -k 5,5n forcing.dat

#Calculate averages for each minute of year
awk -f avg.awk --grp 5 --indx 6 flx_sort_moy.dat > flx_moy.dat
awk -f avg.awk --grp 5 --indx 6 forcing_sort_moy.dat > forcing_moy.dat

#Calculate join keys
awk '{printf("%s-%s-%s ",$6,$4,$5); print $0}' flx_moy.dat > flx_moy_grp.dat
awk '{printf("%s-%s-%s ",$6,$4,$5); print $0}' forcing_moy.dat > forcing_moy_grp.dat

awk '{printf("%s-%s-%s ",$6,$4,$5); print $0}' flx.dat > flx_grp.dat
awk '{printf("%s-%s-%s ",$6,$4,$5); print $0}' forcing.dat > forcing_grp.dat

sort -k 1b,1 flx_moy_grp.dat > flx_moy_grp_srt.dat
sort -k 1b,1 forcing_moy_grp.dat > forcing_moy_grp_srt.dat
sort -k 1b,1 flx_grp.dat > flx_grp_srt.dat
sort -k 1b,1 forcing_grp.dat > forcing_grp_srt.dat


join -v 1 -v 2 -t ' ' flx_grp_srt.dat flx_moy_grp_srt.dat
join -v 1 -v 2 -t ' ' forcing_grp_srt.dat forcing_moy_grp_srt.dat

join -t ' ' flx_grp_srt.dat flx_moy_grp_srt.dat | cut -d" " -f1 --complement > flx_hf_srt.dat
join -t ' ' forcing_grp_srt.dat forcing_moy_grp_srt.dat | cut -d" " -f1 --complement > forcing_hf_srt.dat

sort -o flx_hf.dat -k 1,1n -k 6,6n -k 4,4n -k 5,5n flx_hf_srt.dat
sort -o forcing_hf_srt.dat -k 1,1n -k 6,6n -k 4,4n -k 5,5n forcing_hf.dat

#Sort by day-of-study
#Not needed, as data originally in this format

#Calculate averages/sums for each day
awk -f avg.awk --grp 6 --indx 6 flx.dat > flx_day_avg.dat
awk -f sum.awk --grp 6 --indx 6 flx.dat > flx_day_sum.dat
awk -f avg.awk --grp 6 --indx 6 forcing.dat > forcing_day_avg.dat
awk -f sum.awk --grp 6 --indx 6 forcing.dat > forcing_day_sum.dat
```

### A.3.1 Data aggregation programs

For aggregating data, I created simple *awk* scripts for calculating averages and sums. These implementations aggregate all the variables in the data file grouped by the changes in a selected variable. When this is combined with sorting the rows in data files in specific ways, it is possible to create complex groupings. These also support some rudimentary parameters:

- `grp` controls what column is used for separating the data into different groups. Each time the value in the column changes the aggregation is reset.
- `indx` controls how many columns from the beginning are not aggregated but instead are used in the output as the row identifier for aggregated data.

### Sums

Sums are calculated by a script called `sum.awk`.

sum.awk

```
#
BEGIN {
  old=""
  max_nf=0
  grpby=5
  indx=6
  for (i = 1; i < ARGC; i++) {
    if (ARGV[i] == "--grp"){
      grpby=ARGV[i+1]
      ARGV[i]=""
      ARGV[i+1]=""
    }
    if (ARGV[i] == "--indx"){
      indx=ARGV[i+1]
      ARGV[i]=""
      ARGV[i+1]=""
    }
  }
}
{
  if (max_nf < NF)
    max_nf = NF
  if ($1 == $1 + 0){
    if (old == $grpby)
      for (i = indx+1; i <= NF; i++){
        if ($i == $i + 0){
          values[i] += $i
          counts[i]++
        }
      }
    else{
      if (NR > 2){
        for(i=1; i<=indx;i++){
          printf("%s ", row[i])
        }
        for(i = indx+1; i <= max_nf; i++){
          if (i in values){
            printf("%f ", values[i])
          }
        }
        printf("\n")
      }
      for (i = 1; i <= NF; i++){
        row[i] = $i
      }
      for (i = indx+1; i <= NF; i++){
        if ($i == $i + 0){
          values[i] = $i
          counts[i] = 1
        }
        else {
          values[i]=0
          counts[i]=0
        }
      }
      old = $grpby
    }
  }
  else
    print $0
}
END{
  for(i=1; i<=indx;i++){
    printf("%s ", row[i])
  }
  for(i = indx+1; i <= max_nf; i++){
    if (i in values){
      printf("%f ", values[i])
    }
  }
  printf("\n")
}
```

Set up default values for variables.

Check whether `grp` or `indx` was passed as a variable.

Make sure all columns are calculated.
Check that first column contains numeric data.
If the grouping value has not changed, increment the aggregators.

If the grouping value has changed and this is not the first or second row of the file, print the identifier and aggregated values.

Reset the identifier for aggregated row.

Reset the aggregators.

Store the current grouping value.

If the first column in the data is not numeric, print the whole line.

Print the last identifier and aggregated values.

### Averages

Averages are calculated by `avg.awk` similarly to sums.

avg.awk

```
#
BEGIN {
 old=""
 max_nf=0
 grpby=5
 indx=6
 for (i = 1; i < ARGC; i++) {
  if (ARGV[i] == "--grp"){
   grpby=ARGV[i+1]
   ARGV[i]=""
   ARGV[i+1]=""
  }
  if (ARGV[i] == "--indx"){
   indx=ARGV[i+1]
   ARGV[i]=""
   ARGV[i+1]=""
  }
 }
}
{
 if (max_nf < NF)
  max_nf = NF
 if ($1 == $1 + 0){
  if (old == $grpby)
   for (i = indx+1; i <= NF; i++){
    if ($i == $i + 0){
     values[i] += $i
     counts[i]++
    }
   }
  else{
   if (NR > 2){
    for(i=1; i<=indx;i++){
     printf("%s ", row[i])
    }
    for(i = indx+1; i <= max_nf; i++){
     if (i in values){
      if (counts[i] > 0)
       printf("%f ", values[i]/counts[i])
      else
       printf("NaN ")
     }
    }
    printf("\n")
   }
   for (i = 1; i <= NF; i++){
    row[i] = $i
   }
   for (i = indx+1; i <= NF; i++){
    if ($i == $i + 0){
     values[i] = $i
     counts[i] = 1
    }
    else {
     values[i]=0
     counts[i]=0
    }
   }
   old = $grpby
  }
 }
 else
  print $0
}
END{
 for(i=1; i<=indx;i++){
  printf("%s ", row[i])
 }
 for(i = indx+1; i <= max_nf; i++){
  if (i in values){
   if (counts[i] > 0)
    printf("%f ", values[i]/counts[i])
   else
    printf("NaN ")
  }
 }
 printf("\n")
}
```

Print the average if the variable had any instances, otherwise print the value as missing.

Print the average if the variable had any instances, otherwise print the value as missing.

## A.4 Plotting

Plotting was done using gnuplot scripts. In most cases also calls to any programs outside gnuplot were contained in the scripts.

### A.4.1 Time series

Basic plotting of time series data did not require any tools from *TISEAN*.

daily_avg.gp

```
set terminal epslatex color size 6.5in,5in
set out 'G/NEE_daily_avg.tex'
set multiplot layout 2, 1 rowsfirst downward
set xlabel "$doy$"
set ylabel "$NEE$"
plot 'flx.dat' using ($6+($4+$5/60)/24):7:1 with dots lc
    ↪ palette title "Original data"
set ylabel "$\\overline{NEE}$"
plot 'flx_day_avg.dat' using ($6):($6>355?1/0:$7):1 with
    ↪ lines lc palette title "Daily average"
unset multiplot
set out
```

### A.4.2 Correlations

The autocorrelations were computed using the program `corr` from *TISEAN*.

For the original data these were calculated for both periodic components, the daily and yearly cycles.

corr.gp

```
set terminal epslatex color size 4in,3in
set out 'G/NEE_corr_100.tex'
set xlabel "$\\tau$"
set ylabel "$C(\\tau)$"
set arrow 1 from graph 0, first 0 to graph 1, first 0
    ↪ nohead
set xtics add (13, 13)
set arrow 2 from 13, graph 0 to 13, graph 1 nohead
plot '< corr -c7 flx.dat -D100' with lines linewidth 1
    ↪ linecolor rgbcolor "#000000" title "Autocorrelations
    ↪ of $NEE$"
set out
unset arrow 2
unset xtics
set xtics
set out 'G/NEE_corr_20k.tex'
set xlabel "$\\tau$"
set ylabel "$C(\\tau)$"
```

```
set arrow 1 from graph 0, first 0 to graph 1, first 0 nohead
set arrow 2 from 4380, graph 0 to 4380, graph 1 nohead
set arrow 3 from 8760, graph 0 to 8760, graph 1 nohead
set xrange [0:20000]
set xtics 4380
plot '< corr -c7 flx.dat -D20000' with lines linecolor rgbcolor "#000000" title "Autocorrelations of $NEE$"
set out
unset arrow 2
unset arrow 3
```

For the averaged data only the yearly cycle was left, hence the autocorrelations were calculated only for the longer time frame.

avg_corr.gp

```
set terminal epslatex color size 4in,3in
set out 'G/NEE_avg_corr.tex'
set xlabel "$\\tau$"
set ylabel "$C(\\tau)$"
set arrow 1 from graph 0, first 0 to graph 1, first 0 nohead
set xrange [0:720]
set xtics 200
set xtics add (82, 82)
set arrow 2 from 82, graph 0 to 82, graph 1 nohead
plot '< corr -c7 -D720 flx_day_avg.dat' with lines linewidth 1 linecolor rgbcolor "#000000" title "Autocorrelations of $\\overline{NEE}$"
set out
unset arrow 2
unset xtics
```

### A.4.3 Space time separation

The space time separation plots were produced with the following scripts.

nee_stp.gp

```
set terminal epslatex color size 4in,3in
set out 'G/NEE_stp.tex'
set xlabel "$\\delta t$"
set ylabel "$\\epsilon$"
set xrange [0:144]
plot '< stp -c7 -m2 -d12 -t144 flx.dat' with lines title "Contour lines"
set out
```

avg_stp.gp

```
set terminal epslatex color size 4in,3in
```

```
set out 'G/NEE_avg_stp.tex'
set xlabel "$\\delta t$"
set ylabel "$\\epsilon$"
plot '< stp -c7 -m2 -d91 -t720 flx_day_avg.dat' with lines
    ↪ title "Contour lines"
set out
```

### A.4.4 False nearest neighbors

The graphs for false nearest neighbors were produced with the following script. As the `false_nearest` computer calculates the percentages as a function of dimension, but the plotting would be more natural as a function of the factor $r$ these calculations were done in small batches that were then combined into larger data files.

nee_fnn.gp

```
do for [m=2:7]{
        do for [r=1:8]{
                system sprintf('false_nearest flx.dat -c7 -
                    ↪ m%d -d12 -t480 -M1,%d -f%d -o"flx.
                    ↪ dat_m%d.fnn_f%02d"',m,m,r,m,r)
        }
        system sprintf('cat flx.dat_m%d.fnn_f* > flx.dat_m%
            ↪ d.fnn',m,m)
}
do for [m=8:12]{
        do for [r=1:8:2]{
                system sprintf('false_nearest flx.dat -c7 -
                    ↪ m%d -d12 -t480 -l92040 -M1,%d -f%d -o
                    ↪ "flx.dat_m%d.fnn_f%02d"',m,m,r,m,r)
        }
        system sprintf('cat flx.dat_m%d.fnn_f* > flx.dat_m%
            ↪ d.fnn',m,m)
}
set terminal epslatex color size 4in,3in
set out 'G/NEE_fnn.tex'
set logscale y
set xlabel "Factor $r$"
set ylabel "Fraction of false nearest neighbors"
set arrow 2 from 1, 0.07 to 8, 0.07 nohead
plot for [m=2:7] sprintf('flx.dat_m%d.fnn',m) using ($0+1):2
    ↪ with lines linecolor rgbcolor "#000000" notitle, for[
    ↪ m=8:12] sprintf('flx.dat_m%d.fnn',m) using ($0*2+1):2
    ↪ with lines linecolor rgbcolor "#000000" notitle
set out
```

### A.4.5 Delays

For plotting delays, the program `delay` from *TISEAN* was used to construct the dataset to be plotted.

Delays were plotted by looping for each approximate month (30 day period) into a single multiplot. To speed up the plotting the script calls `delay` on the system level to write the delay vectors to a temporary file.

delay_months.gp

```
system("delay -o -c7 -d12 flx.dat")
set terminal epslatex color size 6.5in,6in
set out "G/NEE_del_months.tex"
set multiplot layout 4, 3 rowsfirst downward
vmarg_l = 1
hmarg_l = 2
vmarg_s = 0.1
hmarg_s = 0.3
set key bottom right
set tmargin vmarg_l
set bmargin vmarg_s
set lmargin hmarg_l
set rmargin hmarg_s
set xtics -30,10,20
set ytics -30,10,20
do for[j=0:11]{
 set xrange [-40:20]
 set yrange [-40:20]
 unset label
 set rmargin (j%3==2 ? hmarg_l : hmarg_s)
 set lmargin (j%3==0 ? hmarg_l : hmarg_s)
 set tmargin (j < 3 ? vmarg_l : vmarg_s)
 set bmargin (j > 8 ? vmarg_l : vmarg_s)
 set format x (j > 8 ? "$%g$" : '')
 set format y (j%3==0 ? "$%g$" : '')
 set label sprintf("$doy \\in [%d,%d]$", (30*j), (30*(j+1))
    ↪ ) at -30, -30
 plot for [i=0:19] "flx.dat.del" every ::(i*365+30*j)*48::(
    ↪ i*365+30*(j+1))*48 with dots linecolor rgbcolor "
    ↪ #000000" notitle
}
unset multiplot
set out
```

The delay plot for $\overline{NEE}$ was broken into months by color to bring a more clear structure to the plot.

avg_delay.gp

```
system("delay -o -c7,2 -M2 -F2,1 -d91 flx_day_avg.dat")
set terminal epslatex color size 4in,3.25in
```

```
set out "G/NEE_avg_delay.tex"
set xlabel "$\\overline{NEE}_n$"
set ylabel "$\\overline{NEE}_{n-\\tau}$"
plot "flx_day_avg.dat.del" using 1:2:3 with points pt 7 ps
    ↪ 0.5 linecolor palette notitle
set out
```